Guidelines on How To run experiments on the SIP botnet detection system

Mohammad AlKurbi
School of Computing Science
Simon Fraser University
Surrey, BC, Canada
mma105@cs.sfu.ca

I. Introduction

This document is a How-To help file. We will describe in it how to run the experiments on the SIP botnet detection system presented in the M.Sc. project report of the detection of botnets mounted on the Session Initiation Protocol. The M.Sc. project report is found under the Resources directory. This How-To complements Chapter 4: Experimental Evaluation, in the previous report. It is based on the testbed setup shown in Figure 1, which takes place in the test network of the Network Systems Lab. The testbed network consists of Ubuntu Linux stations except for the Correlation and Detection engine which is installed on a Windows XP laptop and can be installed anywhere else.

The testbed is consisted of number of components. In order to prepare those components we need to install some external tools such as: MySQL DB [1], Snort [2], Opensips [3], Autosip and Sipbot [4]. We will show how to install each tool, then we will describe step by step how to carry on the experiments on the proposed Correlation and Detection system. What we provide here in this document is just a glimpse of what we do to prepare those tools for the purpose of the experiments, otherwise the full details on how to install the tools and more are available on the tools' references.

The support configuration files are available under Conf directory, and the external tools are available under Tools directory.

II. Installing External Tools

This section provides details on installing or/and initiating the following external tools: MySQL database, Snort, Opensips, Autosip and Sipbot on an Ubuntu Linux OS.

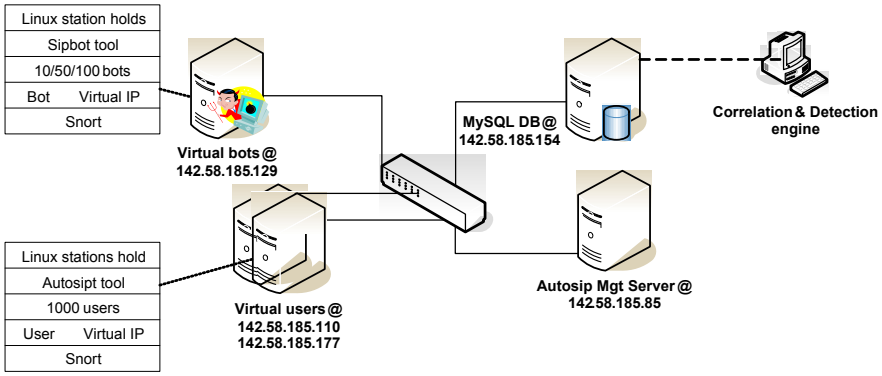

Fig. 1. Testbed Setup

A. *MySQL database*

In order to fulfill the MySQL requirements, we need to install 3 components: MySQL sever [5], Snort database [6], and MySQL Java connector [7]. For full reference on how to install each of these component please follow the previous 3 links. To simplify installing the MySQL server we use the Debian packaging tool, i.e., apt-get install mysql-server. We used it to install the MySQL server version 5.1.37 on 142.58.185.154 station. To install the MySQL Java connector, you need to install it on the same location where the Correlation and Detection engine is installed. You should not need to integrate/adjust the current engine source code to connect to the MySQL database, because it has been done already. You may want to change the address of the destination database though. Any way, the following link provides you with examples on how to use Java to connect to a MySQL database [8]. The following subsection provides more details on how to install Snort which includes setting up the snort database on the MySQL database server.

The following Table I shows few basic and useful MySQL commands:

B. *Snort*

In order to fulfill the Snort [2] requirements, we need to do the following: install Snort on any station that generates SIP traffic, create Snort database, create a local rules file, and edit Snort configuration file. First we need to install Snort that supports the logging feature to a MySQL database. To install Snort use the following command: $ > apt-get install snort-mysql. We installed Snort version 2.7.0 on the the following stations: 142.58.185.110, 142.58.185.177, and 142.58.185.129. Once you install Snort, then create the Snort database following the instructions in one of the following links [6], [9]. The Snort

$ mysql -u root -p#### db_name	Accesses the MySQL server into the specified database as a root user
$ mysqldump -u root -p#### db_name > file_name	Dumps the whole db_name into the specified file
$ mysqldump -u root -p#### db_name < file_name	Restores a dump file into a the specified db
$mysql> create database db_name;	Creates the specified database
$mysql> show tables from db_name;	Lists the tables in the specified database
$mysql> describe data;	Shows the structure of the specified table under the specific database
$mysql> select count(*) from iphdr;	Shows how many entries in the specified table
$mysql> delete * from user;	Deletes all the entries in the specified table
$mysql> mysql –quick;	Helps avoid any memory problems

TABLE I

SOME MYSQL USEFUL COMMANDS.

database schema, i.e., create_mysql file can be found in the following link [10]. The schema is available too under Conf directory. The next step is to create the local rules file which is the file that configure Snort to log the SIP traffic. The local rules files can be found under the same directory as well, and should be transfered into /etc/snort/rules/ directory on the current station. Finally, The snort configuration file, i.e., snort.conf file should be adjusted to do a couple of things, such as defining the home networks, , i.e., HOME_NET variable and configuring the logging to MySQL data base, i.e., output database. The snort configuration file is ready and available under the Conf directory, and need only to be transfered into /etc/snort/ directory.

Assuming that every thing is ready, then to initiate Snort, use the following command: $ > sudo snort -k none -i any -c /etc/snort/snort.conf –process-all-events

C. Opensips

Opensips [3] works as a SIP registrar which is required by the Autosip and Sipbot tools. For simplicity and to avoid any technical problems, we install it on each station that generates SIP traffic. To install Opensips use the following command: $ > apt-get install opensip. We installed Opensips version 1.6.2 on the following stations: 142.58.185.110, 142.58.185.177, and 142.58.185.129. To simplify the testbed setup we choose to configure Opensips to listen only on the station's public IP address. To initiate Opensips use the following command: $ > sudo opensips -l $IP.

D. Autosip

For full documentation about Autosip tool, please refer to [4] and to the tool documentation under the Tools directory. Briefly, this tool is used to emulate the SIP traffic of normal SIP users. The tools consists of two components: a manager and a client. The manager component sets some call parameters, distribute them among the clients, activate some of the clients, then controls the number of active users during time. Autosip is available under the Tools directory [Check]. autosip.c file has been modified to call eXosip_masquerade_contact within init_exosip module. It has been done to set the contact field to the user's IP instead of the Station public IP. This is useful when you have multiple virtual users reside in the same physical machine. Such information is used by the recipient to know where to send back the response.

To install and start using Autosip tool follow these steps:

- Autosip and Sipbot tools require some prior packages. To install pretty much all of them, use the following command:
  $ > apt-get -y –allow-unauthenticated install libexosip2-dev libargtable2-dev libgsl0-dev libboost-filesystem1.35-dev g++ zlib1g-dev bison flex
- Download the package file, uncompress it if it is compressed, and access the tool's home directory, then compile and make the source code by issuing the following command: $ > make all. If succeeded then two binary files are generated: autosip and manager. This step concludes the installation procedure.
- Autosip users use a contacts file to communicate with each other. Autosip comes with support tools, but we wrote another script, aka setup-autosip.net, to help us create the required contacts. To generate the contact file use the following command: $ > setup-autosip.net 10.0 0 1000, which generates 1000 SIP contacts starting from 10.0.0.1
- Before initiating the Autosip clients, we should start the manager by issuing the following command: $ > valgrind manager. valgrind is recommended by the developer which helps preventing the manager from crash. The next thing we have to do is to set the call parameters using the following commands from within the manager prompt:
  - param NUM_FRIENDS 15
  - param CALL_PROBABILITY_FRIEND 0.7
  - param CALL_DURATION_MU 7
  - param CALL_DURATION_STDDEV 4

The default value of μ is 4 and the default value for the standard deviation is 1. Both have been changed to get a better chance of diverse call durations.

  - param NUM_CALLS_PER_HOUR 6

- To initiate the Autosip clients, use the following command: $ > ./autosip -i sip:user$user@$host -r sip:$Station_Public_IP -c $contacts_file -L $User_IP -P 5060 -m $6 –mp=4242
  Where:
  - i: means the SIP identity of the current user.
  - r: the address of the SIP registrar.
  - c: the contact file.
  - L: the address where this user (process) will send out or listen on for incoming SIP sessions.
  - P: the port where this user (process) will listen for incoming SIP sessions on.
  - m: the address of the manager.
  - mp: The manager listening port.
- From the manager prompt, we can check the following:
  - The number of the connected clients by issuing the following command: $< manager >$ status.
  - The settings of the call parameters by issuing the following command: $< manager >$ config.
- To activate the clients and start generating the SIP traffic, use the following command from the manager prompt: execute.

We installed the Autosip manager on 142.58.185.85, and the Autosip client on: 142.58.185.110 and 142.58.185.177. Autosip can be found on those stations under: /home/nsl/Detect-SIP-Botnet/NSL-Eval/autosip/. The support scripts that helps generating SIP traffic can found in: /home/nsl/Detect-SIP-Botnet/NSL-Eval/scripts/.

E. Sipbot

For full documentation about Sipbot tool, please refer to [4] and to the tool documentation under the Tools directory. Briefly, this tool is used to emulate the SIP bots traffic. Sipbot is available under the Tools directory. sipbot.c file has been modified to accept two more additional command line arguments, which are the listening-IP and the listening-Port for the user. These additional arguments control the source IP and the port that the generated SIP traffic comes from. sipbot.c file has been modified as well to call eXosip_masquerade_contact within init_exosip module. We did that to set the contact field to the user's IP instead of the Station public IP. This is useful when you have multiple virtual users reside

in the same physical machine. Such information is used by the recipient to know where to send back the response.

To install and start using Sipbot tool follow these steps:

- Install the prior required packages. Refer to the previous subsection for more information on how to install them.
- Download the package file, uncompress it if it is compressed, and access the tool's home directory. Before compiling the Sipbot source code, you need first to compile the KadC source code. To compile KadC just access its home directory under the Sipbot directory, then issue: make all. Now to compile Sipbot, go out from the KadC directory to the Sipbot tool directory, then issue: make all. If succeeded there will be a sipbot binary file. This step concludes the installation procedure.
- Bots use a contacts file, aka kadc.ini to communicate with each other. Sipbot comes with support tools, but we wrote another script, aka setup-sipbot, to help us create the required contacts. To generate the contact file use the following command: $ > setup-sipbot 10.1.1 1 10, which generates 10 SIP contacts starting from 10.1.1.1
- To initiate a Sipbot user, use the following command: $ > ./sipbot sip:$Registrar $User-IP 5060.

We installed Sipbot on 142.58.185.129, and it can be found under: /home/nsl/Detect-SIP-Botnet/NSL-Eval/sipbot/ directory. The support scripts that helps generating SIP traffic can be found in: /home/nsl/Detect-SIP-Botnet/NSL-Eval/scripts/ directory.

III. Running The experiments

We will show in this section how to carry on experiments on the SIP botnet detection system based on the testbed setup shown in Figure 1. In this testbed setup we have 1000 virtual SIP users, i.e., Autosip users divided into two machines: 142.58.185.110 and 142.58.185.177. The Autosip manager is installed on 142.58.185.85 station. We have a range of of virtual bots, i.e., Sipbot users installed on 142.58.185.129 station. The MySQL server is installed on 142.58.185.154, and the Correlation and Detection engine is installed on an external laptop.

A. Initial steps

Before any experiment, there are some preliminary steps that have to be implemented first. For more information, please refer to Section **??**. The initial steps are as follows:

- At the beginning and for one time only, the necessary external tools have to be installed on the proper locations. External tools include: MySQL DB [1], Snort [2], Opensips [3], Autosip and Sipbot [4].

- At each set of experiments that use the same setting, you need to configure the network and create the contact files as follows:
  - Create the virtual interfaces, but make sure that autosip uses different networks than sipbot:

    Autosip: To generate 1000 virtual interfaces for example do the following
    - sudo ./auto_create_virtual_nics 10.0 0 500 [@ 142.58.185.110]
    - sudo ./auto_create_virtual_nics 10.0 2 500 [@ 142.58.185.177]

      to add 4 C-classes of virtual interfaces starting from 10.0.0.1

    Sipbot: To generate 10 virtual interfaces for example, do as follows
    - sudo ./create_virtual_nics 10.1.1 10 [@ 142.58.185.129]

      to add 10 virtual interfaces starting from 10.1.1.1
  - Create contacts:

    Autosip: To generate 1000 contacts starting from 10.0.0.1 do the following
    - ./setup-autosip.net 10.0 0 1000

    Sipbot: To generate the file of contacts, the initiate-sipbot script calls the setup-sipbot script. To generate 10 contacts starting from 10.1.1.1, the setup-sipbot script is called in this format:
    - setup-sipbot 10.1.1 1 10
- Before running each experiment, stop any running process and do the following:
  - Backup the database content of the previous experiment -if any-, then clear its content if you wish.
  - Start Snort on each station that generates a SIP traffic. if you choose to log to a different database, then make sure that Snort is configured properly to log to the designated data base.
  - Start Autosip manager, then set the Autosip Call parameters.

B. Start the experiment

Once you prepare the database, start the Autosip manager, set the call parameters, and start the snort on all stations that generate SIP traffic, then you are ready to start the actual experiment. To start the experiment do the following:

- Start Autosip users as follows:
  - ./initiate-autosip.net 10.0 0 5060 500 142.58.185.110 142.58.185.85 contacts.txt
  - ./initiate-autosip.net 10.0 2 5060 500 142.58.185.177 142.58.185.85 contacts.txt

  - You may use the status command from the Autosip manager prompt to check that the number of users is what you expect. Otherwise debug the case.
  - Start generating the SIP traffic of the Autosip users by issuing the execute command from the Autosip manager prompt, which will activate the users.
- Start Sipbot users as follows:
  - ./initiate-sipbot 10.1.1 1 5060 10 142.58.185.129
  - Some sipbot processes might crash during time. Therefore, you may add a cron job that make sure that does not happen, and if it does to correct the situation.

IV. The Correlation and Detection engine

The source code of the Correlation and Detection engine is a self-content, and there is no need for further guidelines on how to run it especially when the online mode or the incremental is selected. To be able to evaluate the same data multiple times for different settings the offline mode has to be selected. At any mode the system generates a statistics file that includes all the information needed to evaluate the engine precision and its time cost. We wrote a number of MS-DOS scripts, such as: auto-eval-mw.bat and auto-eval-m-sw.bat to automate the evaluation process as much as possible when the offline and non-incremental modes are selected. The MS-DOS scripts are found under the Tools directory in the Correlation and Detection engine directory. Before showing how to use those scripts, it is good to give an overview of the final versions of the source code. The version 5.5 is the one used to evaluate the precision of the proposed system, because it was the latest version at that time. The version 7.5 is the final and the latest version of the proposed system, which has additional features, but the core engine is almost identical with the version 5.5. Although version 7.5 and 5.5 seem identical, we noticed through preliminary testing that β has to be sat to 0.78 instead of 0.8 in order for the proposed system at the incremental mode to work as expected. To evaluate the Correlation and Detection engine we provide the following examples noting that 10.0 represents the basis network for the normal users, 10.1 is the basic network for the bots, α= 0.05, β= 0.8, and the mode has to be always sat to offline:

- To evaluate the engine for 24 hours ($\frac{60}{15} \times 24 = 96$) with a 3 hour time window W, and a 15 minutes sliding window S, then run the following command:
  $\$ >$ auto-eval-mw.bat 180 15 96 10.0 10.1 0.05 0.8 false
- To evaluate the engine for 6 sliding window S sizes: 5min, 10min, 15min, 20min, 25min, and 30min, then run the following command:
  $\$ >$ auto-eval-m-sw.bat 120 5 24 10.0 10.1 0.05 0.8 false 5 6

We evaluate each S for 24 hours with a 2 hours time window W, and an initial value of S as 5 minutes that increments by 5 minutes.

V. Conclusion

We showed in this guidelines how to setup the testbed, install the required tools, run the experiments, and finally how to use the proposed system to evaluate its precision and time cost.

References

[1] "MySQL: The world's most popular open source database software," http://dev.mysql.com/.

[2] "Snort: Open source network intrusion prevention and detection system," http://www.snort.org/.

[3] "Opensips: Open source implementation of SIP server," http://opensips.org/.

[4] A. Berger and M. Hefeeda, "Exploiting SIP for botnet communication," in *Proc. of Secure Network Protocols Workshop (NPSec'09)*, Princeton, NJ, October 2009.

[5] "MySQL: The world's most popular open source database software," http://dev.mysql.com/downloads/mysql/.

[6] "Snort: Open source network intrusion prevention and detection system," http://www.snort.org/assets/167/deb_snort_howto.pdf.

[7] Http://dev.mysql.com/usingmysql/java/.

[8] Http://dev.mysql.com/doc/refman/5.1/en/connector-j-examples.html.

[9] "Snort: Open source network intrusion prevention and detection system," http://ubuntuforums.org/showthread.php?t=483488.

[10] "Snort: Open source network intrusion prevention and detection system," http://cvs.snort.org/viewcvs.cgi/snort/schemas/.